

**AIAA Space and Astronautics Forum**  
**Session IS-01 ISHM for Space Systems**  
**September 13, 2016**

**Functional Fault Model Development Process to Support  
Design Analysis and Operational Assessment**

**Kevin J. Melcher**

NASA Glenn Research Center  
Cleveland, Ohio 44135 U.S.A

**William A. Maul**

Vantage Partners LLC.  
Brookpark, OH 44142 U.S.A.

**Joseph A. Hemminger**

ZIN Technologies, Inc.  
Brookpark, OH 44142 U.S.A.



# Presentation Outline

- Introduction
  - Purpose
  - Motivation
  - What is a Functional Fault Model
- FFM Development Process
  - Phase 1: Knowledge Acquisition
  - Phase 2: Conceptual Design
  - Phase 3: Implementation & Verification
  - Phase 4: Application
- Concluding Remarks



# Introduction

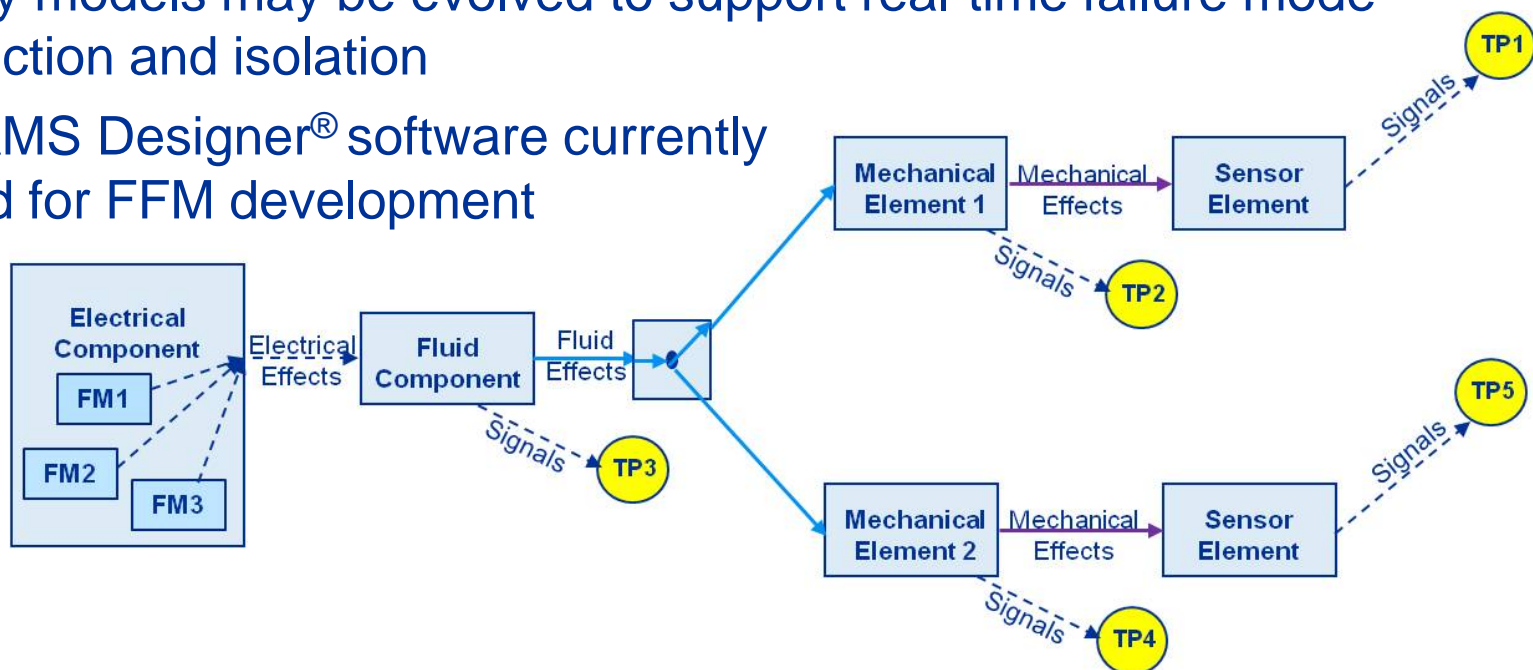
- Purpose of paper
  - To characterize and document the current process used by NASA to develop functional fault models (FFMs)
  - To identify new technologies and capabilities that contribute to an improved process.
- Motivation for the paper
  - Process has evolved over past 10 years with push to support development of new NASA human-rated space systems
  - Modeling guidelines, best practices, and software tools have been developed to substantially improve:
    - The efficiency of the FFM development and verification process
    - The utility and impact of FFM applications
  - Benchmark for future FFM development efforts as the process continues to evolve



## Introduction

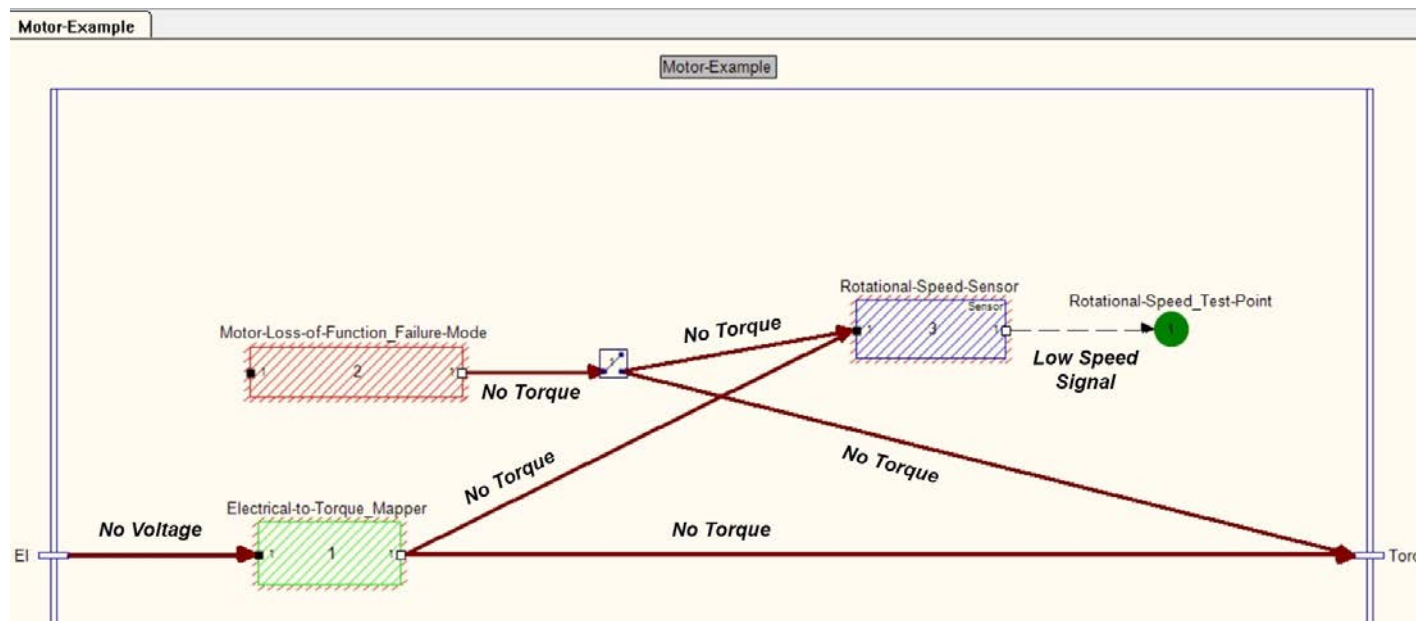
# What is a Functional Fault Model?

- Directed graph representation of failure effect propagation paths within the system architecture
- Developed to address limitations of traditional methods
- Initial models can be qualitative supporting requirements verification early in system design process
- Early models may be evolved to support real-time failure mode detection and isolation
- TEAMS Designer<sup>®</sup> software currently used for FFM development



## Introduction

# Functional Fault Models in TEAMS

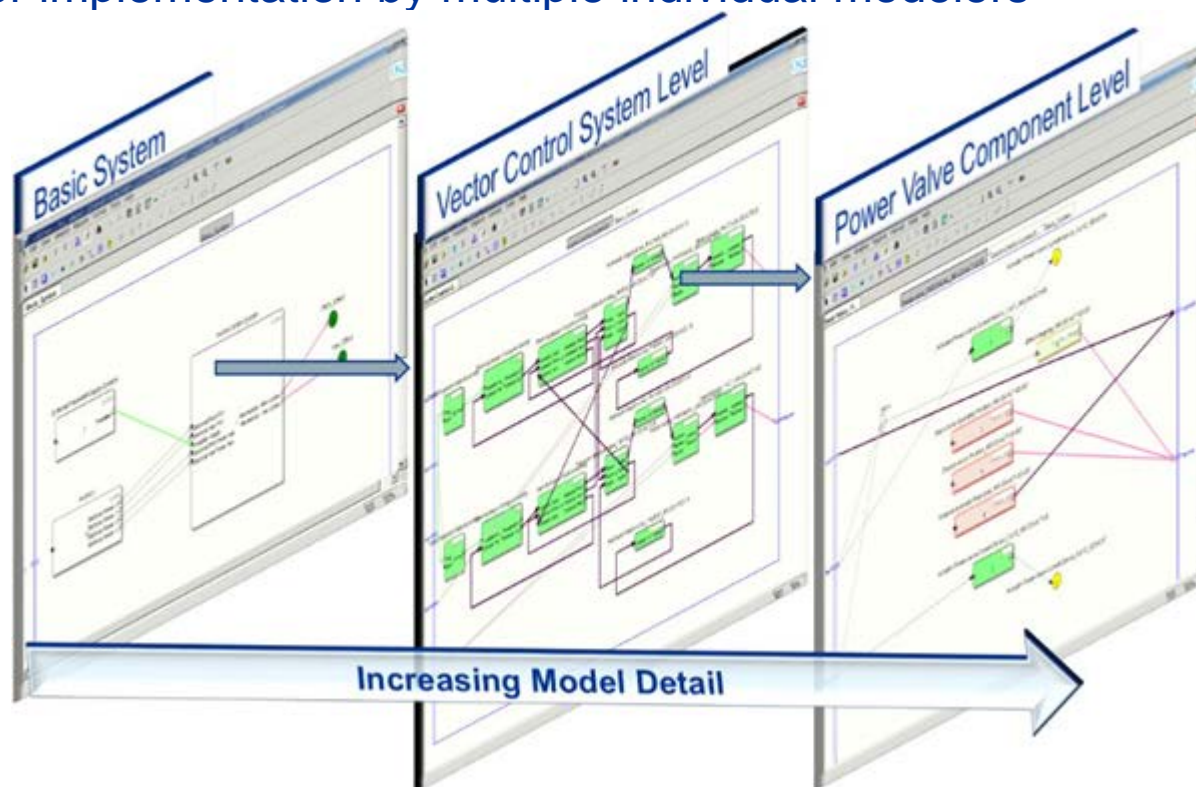


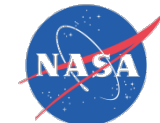
- Modules – represent systems, assemblies, components
- Failure Mode Modules – contain the qualitative failure information
- Mapper Modules – represent the nominal transition of the failure effects being propagated
- Test-Points – represent the observation points of the system (typically associated with sensors)
- Tests – detect specific failure effects

## Introduction

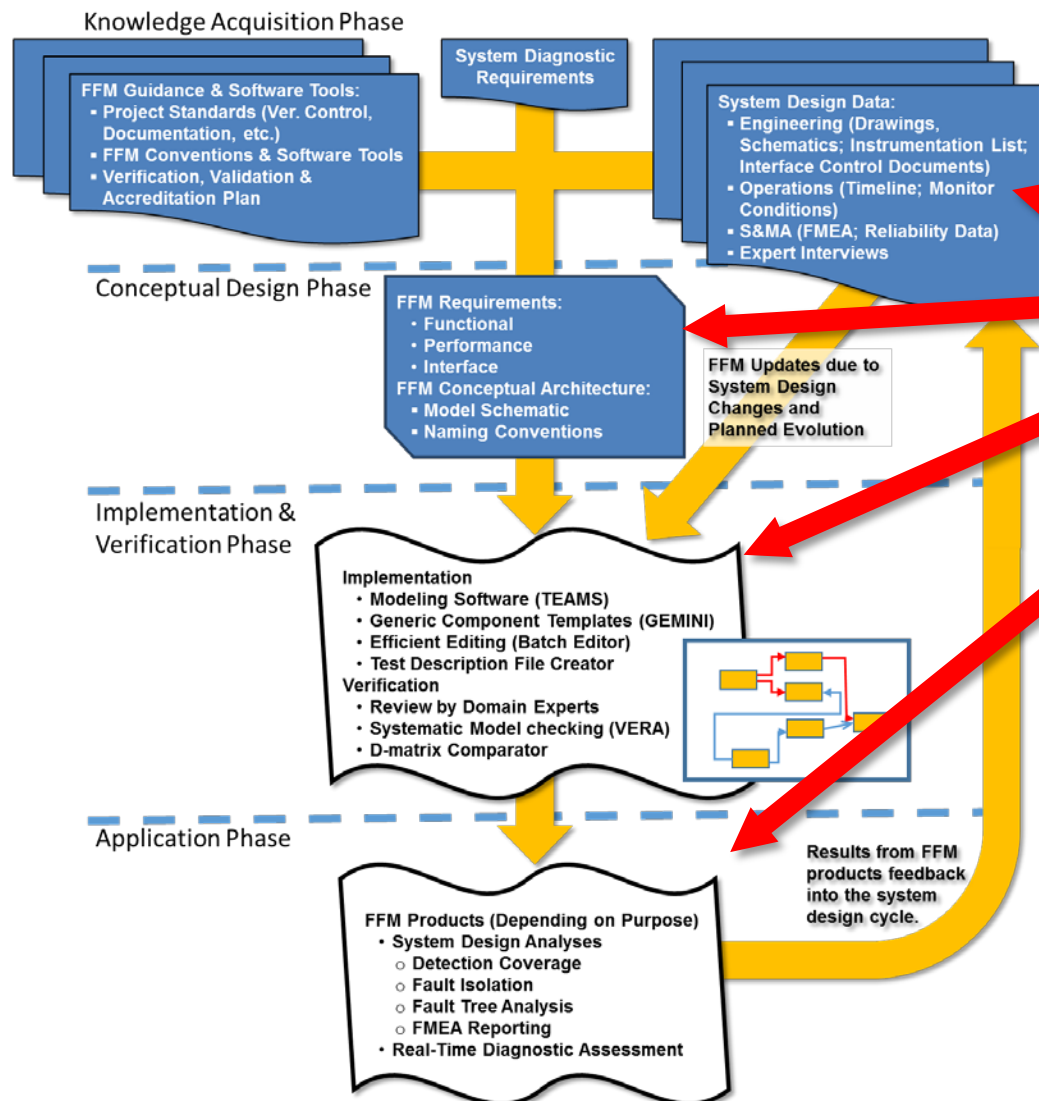
# Functional Fault Models in TEAMS

- Hierarchical modeling capability:
  - Supports a model structure that reflects a hierarchical decomposition of system hardware & software
  - Facilitates portioning of large complex models into smaller models for implementation by multiple individual modelers



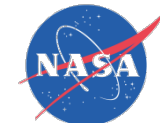


# FFM Development Process: Overview

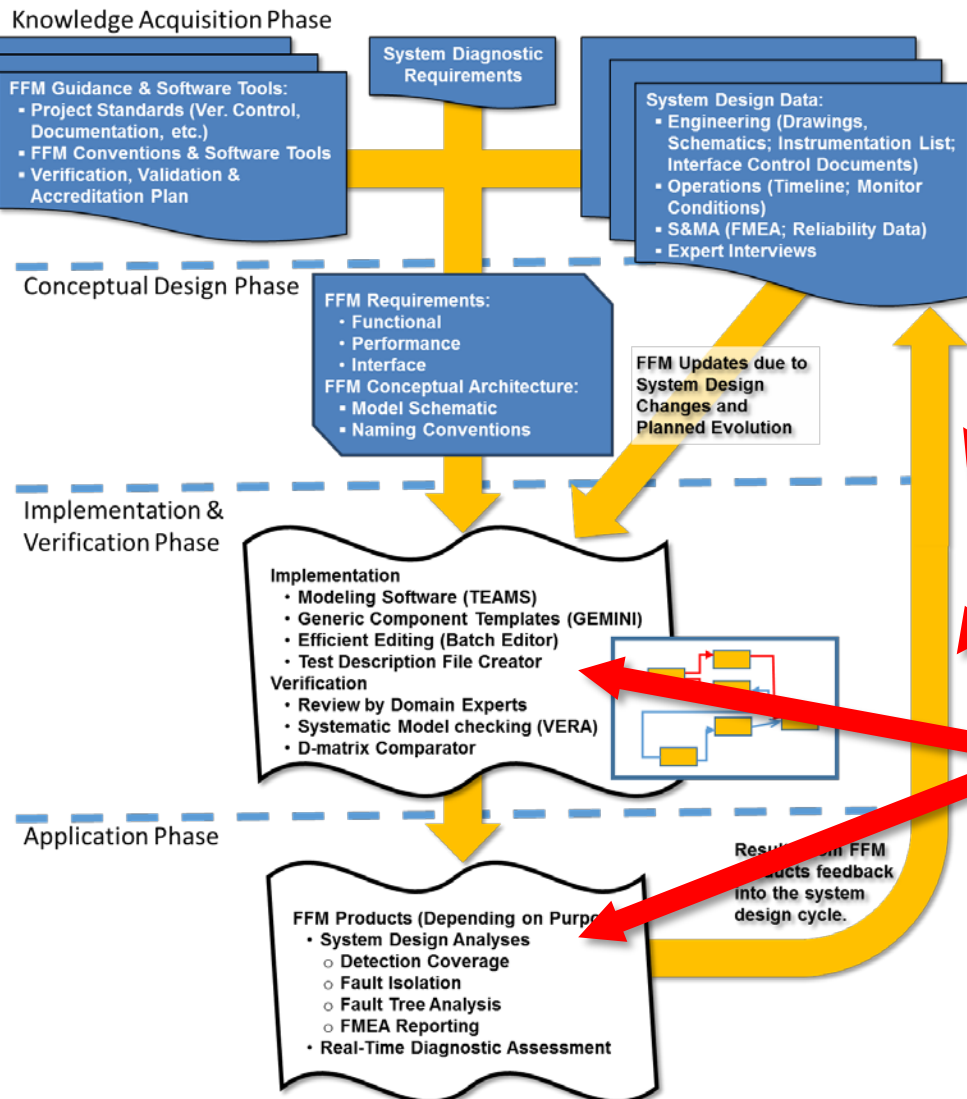


- High-level traditional modeling process with four (4) phases:
  1. Knowledge Acquisition
  2. Conceptual Design
  3. Implementation and Verification
  4. Application





# FFM Development Process: Overview



- High-level traditional modeling process with four (4) phases:

1. Knowledge Acquisition
2. Conceptual Design
3. Implementation and Verification
4. Application

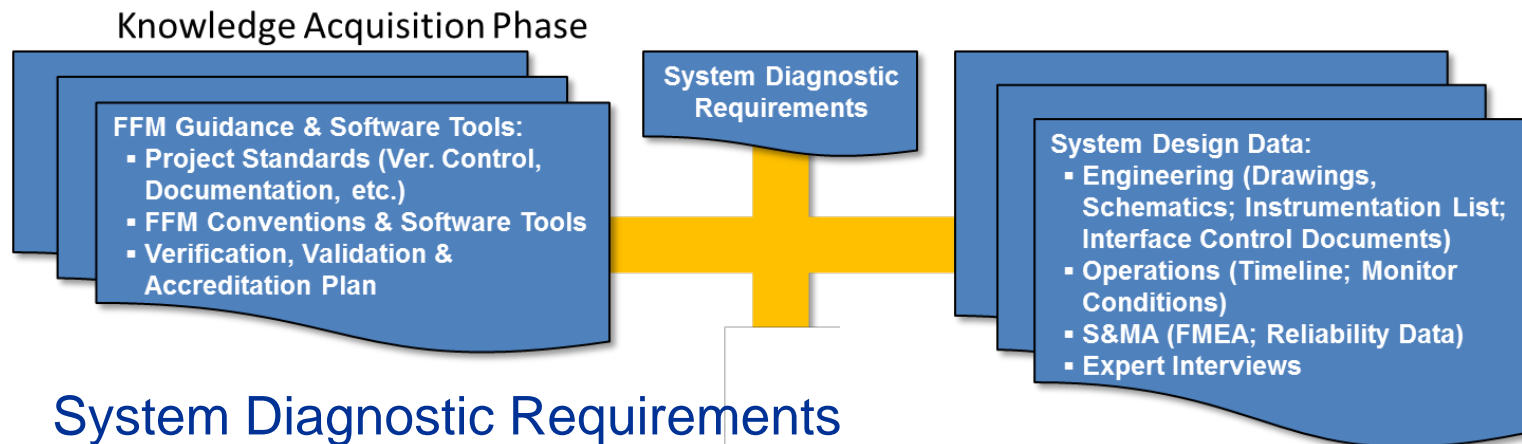
Ideally, analysis results from the *Application Phase* feed back into the system design

Each phase incorporates FFM-specific features

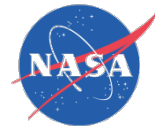




# Phase 1: Knowledge Acquisition



- **System Diagnostic Requirements**
  - System-level requirements impact all phases of the FFM dev. process
  - Examples: Abort conditions, launch commit criteria, line replaceable units
- **System Design Data**
  - Information that defines the system design and operation
  - Examples: Engineering drawings and reports, concept of operations, Failure Modes and Effects Analysis (FMEA)
- **FFM Guidance & Software Tools**
  - Information needed to implement a model that informs the Conceptual Design, Implementation & Verification, and Application phases
  - Examples: modeling conventions, model VV&A plan
- **Establish System Breakdown Structure (SRS) & other databases**



# Phase 2: FFM Conceptual Design

Conceptual Design Phase

FFM Requirements:

- Functional
- Performance
- Interface

FFM Conceptual Architecture:

- Model Schematic
- Naming Conventions

- FFM Requirements

- Flowed down from system diagnostic requirements, FFM conventions and practices
- Functional: Failure modes, test points, test logic
- Performance: Time to detect/isolate failures, False positive/negative rates

- Interface: FFM-to-FFM, system to FFM to key decision makers on ground or vehicle

- FFM Conceptual Architecture

- System Operational Profile
- Model Schematic/Structure
- Naming Conventions



## Phase 2: Conceptual Design

# FFM Rqmts: Modeling Conventions & Practices

- Approved by NASA's SLS, Orion, and Ground Systems FFM communities.
- Documents FFM best practices of all three communities.
- Benefits:
  - Model elements and sub-models have consistent look and feel
  - Improves human understanding
  - Enables more efficient integration of independently developed FFMs
  - Facilitates development of the interfaces needed for integration of FFMs with real-time systems
  - Improves traceability of model features back to source documents





# Phase 3: Implementation & Verification

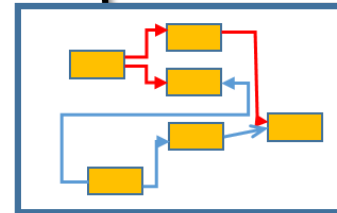
Implementation &  
Verification Phase

## Implementation

- Modeling Software (TEAMS)
- Generic Component Templates (GEMINI)
- Efficient Editing (Batch Editor)
- Test Description File Creator

## Verification

- Review by Domain Experts
- Systematic Model checking (VERA)
- D-matrix Comparator



Start Modeling

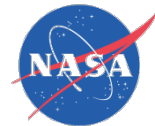
## Create/Revise FFM

- TEAMS Designer software
- NASA-developed software tools
  - Batch Editor
  - GEneric Model INstantiator (GEMINI) software
  - Test Description File Creator

## Verify FFM

- Review by domain experts
- NASA-developed software tools
  - VERification Analysis (VERA) Tool
  - D-matrix Comparator

Tightly Coupled Lower-Level Processes



## Phase 3: Implementation & Verification

# NASA-Developed Tools

- Batch Editor
  - Includes model query and update commands to efficiently make broad systematic changes to FFMs (not available in TEAMS)
  - Commands cover a wide variety of TEAMS modules & features
  - Graphical or command line user interfaces
  - Used by several other NASA-developed FFM tools to extract model information
- GEneric Model INstantItation (GEMINI) Tool
  - Supports the use of generic model libraries
  - Generates component-specific FFMs by adding user-provided component data to generic component models
- Test Description File Creator
  - Aligns real-time system data to FFM tests (mode dependent)
  - Defines thresholds—results in quantitative diagnostic assessment



## Phase 3: Implementation & Verification

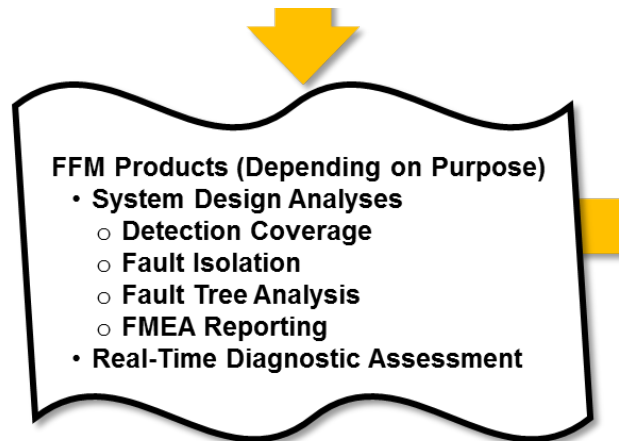
# NASA-Developed Tools

- VERification Analysis (VERA) Tool
  - Checks model for adherence to NASA FFM conventions & practices
  - Reads model information into MS Excel Workbook
  - Analyzes model in four areas:
    - Technical
    - Practices & Conventions
    - Cosmetic
    - Informational content
  - Generates detailed reports that identify non-compliant FFM features
  - Provides scores to support accreditation of the model for operational use.
- D-Matrix Comparator
  - Reports differences between D-matrices from two different FFMs
  - Useful for regression testing to ensure minor model checks reflected in results



# Phase 4: Application

Application Phase



## FFM Analysis Products

- **Failure Detectability Report**
  - Analyzes FFM for detected / undetected failure modes
  - Verifies detection coverage rqmts.
- **Test Utilization Report**
  - Analyzes FFM for used/not used tests (sensors)
  - Supports sensor selection/buy-in
- **Fault Isolation Report**
  - Analyzes failure mode uniqueness / ambiguity
  - Verifies rqmts for algorithms used to detect failure effects
- **Component Isolation Report**
  - Analyzes isolation of failure modes to user-defined components
  - Verifies requirements for line replaceable units
- **FMEA Report**
  - Uses data embodied in FFM to generate a report containing failure mode description data and detection capabilities from FFM





# Phase 4: Application

Application Phase

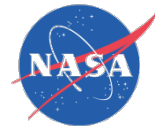


## Moving FFM from analytical use to real-time:

- Interface policies & software for generating FFM input from the real-time data
  - Handling dynamic data
  - Loss of data
  - Align FFM tests with software that processes real-time data

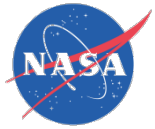
## Real-time Diagnostic Assessment:

- Provide a list of failure modes, components, and sensors that align to the latest test detection
- Textual information traceable to design and FFM documentation
- Used by decision makers in flight and on ground.



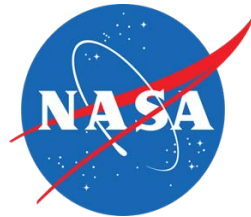
# Concluding Remarks

- This paper presented an iterative, four (4) phase process to support the development of FFMs.
- Special emphasis was placed on key approaches, capabilities, and tools that are unique to FFMs.
- The process has proved beneficial to recent systems engineering assessments under NASA's Ares I, Space Launch System, and Ground Systems Development and Operations Programs.
- Continued evolution of the process is anticipated as:
  - Current capabilities mature,
  - Additional capabilities are developed,
  - All capabilities are demonstrated in future flight and ground systems.



# Acknowledgements

This work was conducted under the  
NASA Space Launch System Program  
Mission and Fault Management Project



## Presenter Contact Information

Email: [kevin.j.melcher@nasa.gov](mailto:kevin.j.melcher@nasa.gov)

Office Phone: 216-433-3743